CLAIMS:

1. A method for calculating a branch target address comprising the steps of:

fetching a branch instruction from a memory, wherein said branch instruction stores an offset of a target address comprising n bits;

calculating n-1 least significant bits of said target address of said branch instruction; and

replacing n-1 least significant bits of said offset of said target address with said n-1 least significant bits of said target address of said branch instruction.

2. The method as recited in claim 1 further comprises the step of:

appending a carry bit to said branch instruction thereby increasing a length of said branch instruction by one bit.

3. The method as recited in claim 2 further comprising the steps of:

calculating a set of upper order bit value combinations of an address of said branch instruction;

storing said branch instruction storing said n-1 least significant bits of said target address in a cache;

retrieving said branch instruction storing said n-1 least significant bits of said target address from said cache; and

selecting a one of said set of upper order bit value combinations of said address of said branch instruction.

4. The method as recited in claim 3 further comprising the step of:

appending said selected one of said set of upper order bit value combinations of said address of said branch instruction with said n-1 least significant bits of said target address to calculate said target address of said branch instruction.

5. The method as recited in claim 1, wherein said step of calculating n-1 least significant bits of said target address of said branch instruction comprises the step of:

adding a value stored in said n-1 least significant bits of said offset of said target address stored in said branch instruction with a value stored in said n-1 least significant bits of said address of said branch instruction.

6. The method as recited in claim 3, wherein said set of upper order bit value combinations of said address of said branch instruction comprises one or more of the following: a value in said upper order bits of said address of said branch instruction incremented by one, said value in said upper order bits of said address of said branch instruction decremented by one and said value in said upper order bits of said address of said branch instruction.

7. The method as recited in claim 3, wherein said one of said set of upper order bit value combinations is selected in response to a value in a sign bit and a value in said carry bit in said branch instruction.

8. A system, comprising:

a memory configured to store instructions;

a cache coupled to said memory, wherein said cache is configured to fetch an instruction from said memory; and

an encoding logic unit coupled to said cache, wherein said encoding logic unit is configured to encode said fetched instruction, wherein said encoding logic unit is configured to determine if said instruction is a relative branch instruction, wherein said relative branch instruction stores an offset of a target address comprising n bits, wherein if said instruction is said relative branch instruction then said encoding logic unit is configured to calculate n-1 least significant bits of said target address, wherein said encoding logic unit is further configured to replace n-1 least significant bits of said offset of said target address with said n-1 least significant bits of said target address.

9. The system as recited in claim 8, wherein said encoding logic unit is further configured to append a carry bit to said relative branch instruction thereby increasing a length of said relative branch instruction by one bit.

10. The system as recited in claim 9 further comprises:

a fetch unit coupled to said cache, wherein said fetch unit is configured to calculate a set of upper order bit value combinations of an address of said relative branch instruction.

11. The system as recited in claim 10, wherein said cache is configured to store said relative branch instruction storing said n-1 least significant bits of said target address.

12. The system as recited in claim 11 further comprises:

a logic unit coupled to said cache, wherein said logic unit is configured to retrieve said relative branch instruction storing said n-1 least significant bits of said target address from said cache.

13. The system as recited in claim 12, wherein said logic unit is further configured to receive said set of upper order bit value combinations of said address of said relative branch instruction from said fetch unit.

14. The system as recited in claim 13, wherein said logic unit is further configured to select a one of said set of upper order bit value combinations of said address of said relative branch instruction.

15. The system as recited in claim 14, wherein said logic unit is further configured to append said selected one of said set of upper order bit value combinations of said address of said relative branch instruction with said n-1 least significant bits of said target address to calculate said target address.

16. The system as recited in claim 8, wherein said encoding logic unit is configured to calculate said n-1 least significant bits of said target address by adding a value stored in said n-1 least significant bits of said offset of said target address stored in said relative branch instruction with a value stored in said n-1 least significant bits of said address of said relative branch instruction.

17. The system as recited in claim 10, wherein said set of upper order bit value combinations of said address of said relative branch instruction comprises one or more of the following: a value in said upper order bits of said address of said relative branch instruction incremented by one, said value in said upper order bits of said address of said relative branch instruction decremented by one and said value in said upper order bits of said address of said relative branch instruction.

18. The system as recited in claim 14, wherein said one of said set of upper order bit value combinations is selected in response to a value in a sign bit and a value in said carry bit in said branch instruction.

19. A system, comprising:

means for storing instructions;

means for fetching an instruction;

means for encoding said fetched instruction; and

means for determining if said instruction is a relative branch instruction, wherein said relative branch instruction stores an offset of a target address comprising n bits, wherein if said instruction is said relative branch instruction then the system further comprises:

means for calculating n-1 least significant bits of said target address; and

means for replacing n-1 least significant bits of said offset of said target address with said n-1 least significant bits of said target address.

20. The system as recited in claim 19 further comprises:

means for appending a carry bit to said relative branch instruction thereby increasing a length of said relative branch instruction by one bit.

21. The system as recited in claim 20 further comprises:

means for calculating a set of upper order bit value combinations of an address of said relative branch instruction.

22. The system as recited in claim 21 further comprises:

means for storing said relative branch instruction storing said n-1 least significant bits of said target address.

23. The system as recited in claim 22 further comprises:

means for retrieving said relative branch instruction storing said n-1 least significant bits of said target address.

24. The system as recited in claim 23 further comprises:

means for receiving said set of upper order bit value combinations of said address of said relative branch instruction.

25. The system as recited in claim 24 further comprises:

means for selecting a one of said set of upper order bit value combination of said address of said relative branch instruction.

26. The system as recited in claim 25 further comprises:

means for appending said selected one of said set of upper order bit value combination of said address of said relative branch instruction with said n-1 least significant bits of said target address to calculate said target address.

27. The system as recited in claim 19, wherein said n-1 least significant bits of said target address is calculated by adding a value stored in said n-1 least significant bits of said offset of said target address stored in said relative branch instruction with a value stored in said n-1 least significant bits of said address of said relative branch instruction.

28. The system as recited in claim 21, wherein said set of upper order bit value combinations of said address of said relative branch instruction comprises one or more of the following: a value in said upper order bits of said address of said relative branch instruction incremented by one, said value in said upper order bits of said address of said relative branch instruction decremented by one and said value in said upper order bits of said address of said relative branch instruction.

29. The system as recited in claim 25, wherein said one of said set of upper order bit value combinations is selected in response to a value in a sign bit and a value in said carry bit in said branch instruction.

30. A processor, comprising:

a cache configured to fetch an instruction; and

an encoding logic unit coupled to said cache configured to encode said fetched instruction, wherein said encoding logic unit is configured to determine if said instruction is a relative branch instruction, wherein said relative branch instruction stores an offset of a target address comprising n bits, wherein if said instruction is said relative branch instruction then said encoding logic unit is configured to calculate n-1 least significant bits of said target address, wherein said encoding logic unit is further configured to replace n-1 least significant bits of said offset of said target address with said n-1 least significant bits of said target address.

31. The processor as recited in claim 30, wherein said encoding logic unit is further configured to append a carry bit to said relative branch instruction thereby increasing a length of said relative branch instruction by one bit.

32. The processor as recited in claim 31 further comprises:

a fetch unit coupled to said cache, wherein said fetch unit is configured to calculate a set of upper order bit value combinations of an address of said relative branch instruction.

33. The processor as recited in claim 32, wherein said cache is configured to store said relative branch instruction storing said n-1 least significant bits of said target address.

34. The processor as recited in claim 33 further comprises:

a logic unit coupled to said cache, wherein said logic unit is configured to retrieve said relative branch instruction storing said n-1 least significant bits of said target address from said cache.

35. The processor as recited in claim 34, wherein said logic unit is further configured to receive said set of upper order bit value combinations of said address of said relative branch instruction from said fetch unit.

36. The processor as recited in claim 35, wherein said logic unit is further configured to select a one of said set of upper order bit value combinations of said address of said relative branch instruction.

37. The processor as recited in claim 36, wherein said logic unit is further configured to append said selected one of said set of upper order bit value combinations of said address of said relative branch instruction with said n-1 least significant bits of said target address to calculate said target address.

38. The processor as recited in claim 30, wherein said encoding logic unit is configured to calculate said n-1 least significant bits of said target address by adding a value stored in said n-1 least significant bits of said offset of said target address stored in said relative branch instruction with a value stored in said n-1 least significant bits of said address of said relative branch instruction.

39. The processor as recited in claim 32, wherein said set of upper order bit value combinations of said address of said relative branch instruction comprises one or more of the following: a value in said upper order bits of said address of said relative branch instruction incremented by one, said value in said upper order bits of said address of said relative branch instruction decremented by one and said value in said upper order bits of said address of said relative branch instruction.

40. The processor as recited in claim 36, wherein said one of said set of upper order bit value combinations is selected in response to a value in a sign bit and a value in said carry bit in said branch instruction.

41. A processor, comprising:

means for fetching an instruction; and

means for determining if said instruction is a relative branch instruction, wherein said relative branch instruction stores an offset of a target address comprising n bits, wherein if said instruction is said relative branch instruction then the processor further comprises:

means for calculating n-1 least significant bits of said target address; and

means for replacing n-1 least significant bits of said offset of said target address with said n-1 least significant bits of said target address.

42. The processor as recited in claim 41 further comprises:

means for appending a carry bit to said relative branch instruction thereby increasing a length of said relative branch instruction by one bit.

43. The processor as recited in claim 42 further comprises:

means for calculating a set of upper order bit value combinations of an address of said relative branch instruction.

44. The processor as recited in claim 43 further comprises:

means for storing said relative branch instruction storing said n-1 least significant bits of said target address.

45. The processor as recited in claim 44 further comprises:

means for retrieving said relative branch instruction storing said n-1 least significant bits of said target address.

46. The processor as recited in claim 45 further comprises:

means for receiving said set of upper order bit value combinations of said address of said relative branch instruction from said fetch unit.

47. The processor as recited in claim 46 further comprises:

means for selecting one of said set of upper order bit value combinations of said address of said relative branch instruction.

48. The processor as recited in claim 47 further comprises:

means for appending said selected one of said set of upper order bit value combinations of said address of said relative branch instruction with said n-1 least significant bits of said target address to calculate said target address.

49. The processor as recited in claim 41 further comprises:

means for calculating said n-1 least significant bits of said target address by adding a value stored in said n-1 least significant bits of said offset of said target address stored in said relative branch instruction with a value stored in said n-1 least significant bits of said address of said relative branch instruction.

50. The processor as recited in claim 43, wherein said set of upper order bit value combinations of said address of said relative branch instruction comprises one or more of the following: a value in said upper order bits of said address of said relative branch instruction incremented by one, said value in said upper order bits of said address of said relative branch instruction decremented by one and said value in said upper order bits of said address of said relative branch instruction.

51. The processor as recited in claim 47, wherein said one of said set of upper order bit value combinations is selected in response to a value in a sign bit and a value in said carry bit in said branch instruction.